Peer to Peer Networks and Web Services for a Community Grid

PTLIU Laboratory for Community Grids

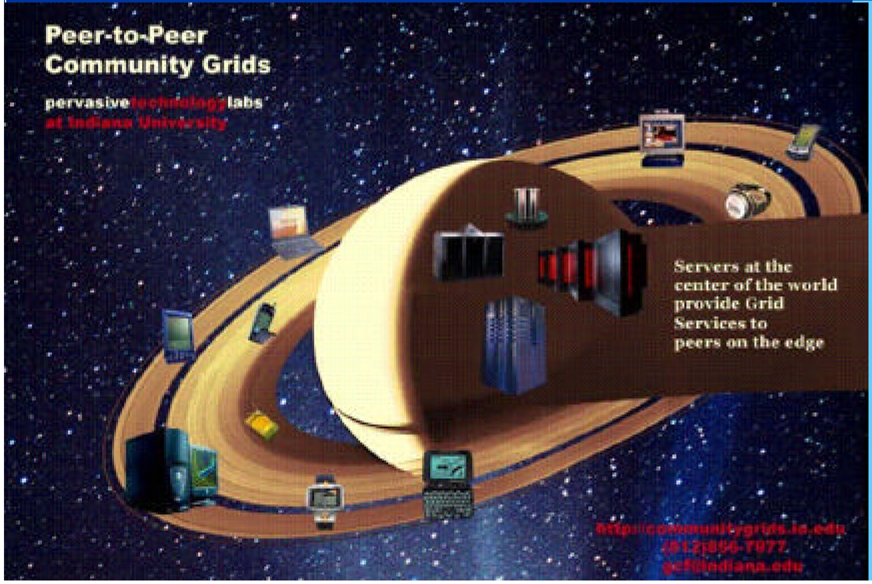
Geoffrey Fox, Marlon Pierce, Shrideep Pallickara, Choonhan Youn

Computer Science, Informatics, Physics

Indiana University

Bloomington IN 47404

gcf@indiana.edu


Peer-to-Peer
Community Grids
pervasivetechnologylabs
at Indiana University
Servers at the center of the world provide Grid Services to peers on the edge
http://communitygrids.iu.edu
(812)856-7977
gcf@indiana.edu

pervasivetechnologylabs

AT INDIANA UNIVERSITY

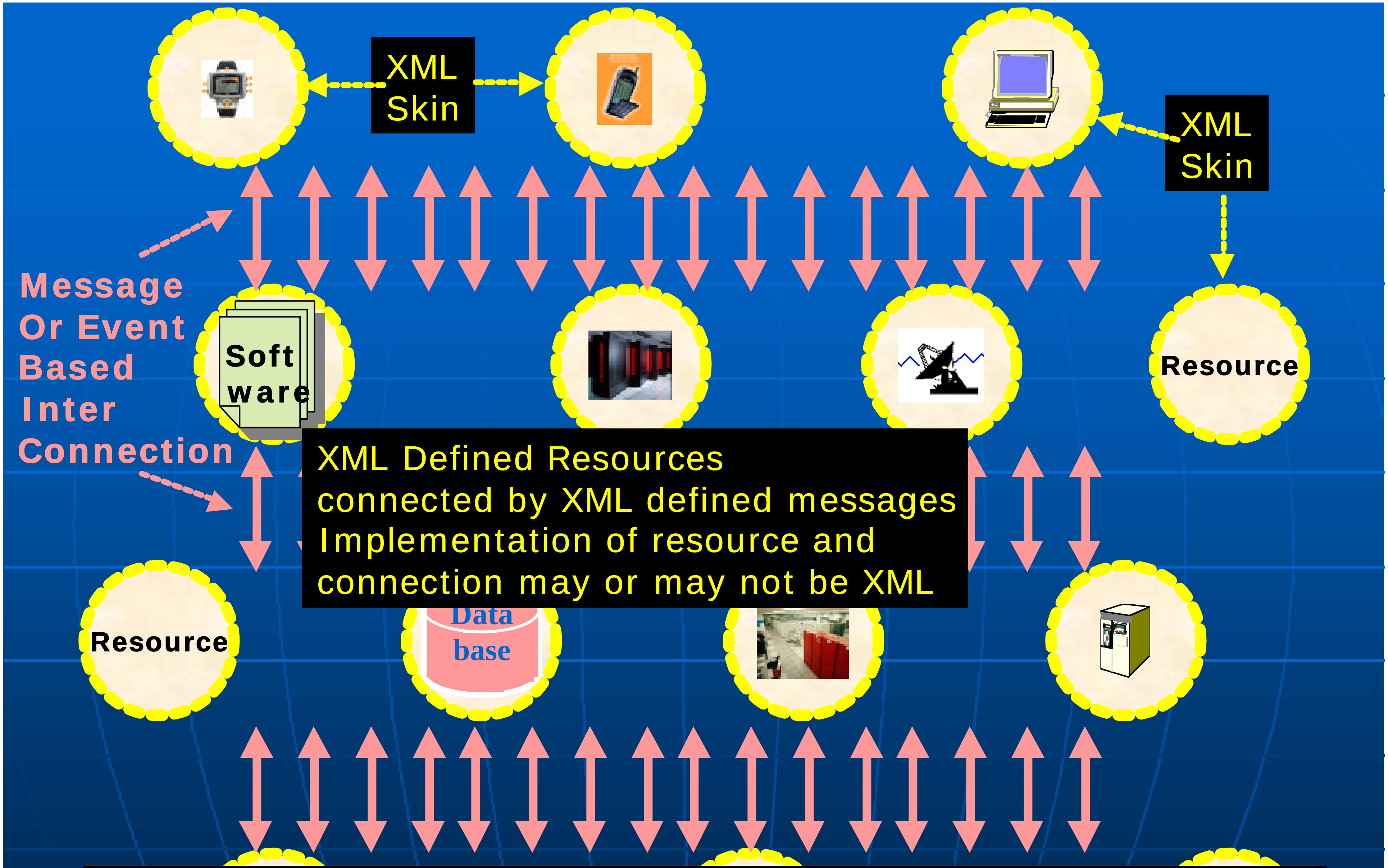
P2P Grid Architecture I

- "Everything electronic" is a resource
  - Computers
  - Programs
  - Data (from sensors to this presentation to email to databases)
  - People
- Resources are labeled by XML
  - URI from URL (location) to URN (property tag)
  - Metadata
  - Software Interfaces
  - Personal Information
- XML Interfaces may be "virtual"
  - Define in XML but "compile" to optimized form for performance functionality accessibility trade-offs

P2P Grid Architecture II

- Nearly all resources have a web interface
  - Including people and software components
  - All resources have natural GUI from browser
- Everything is an Object (as opposed to or in addition to being a table or an array)
- Objects have well defined interfaces which can and should be standardized
- Essentially all resources connect with messages which must also have a possibly virtual XML specification
  - This includes resources (such as functions) running in same memory space
  - As well as the more obvious coarser grain web applications

Peer to Peer Community Grid

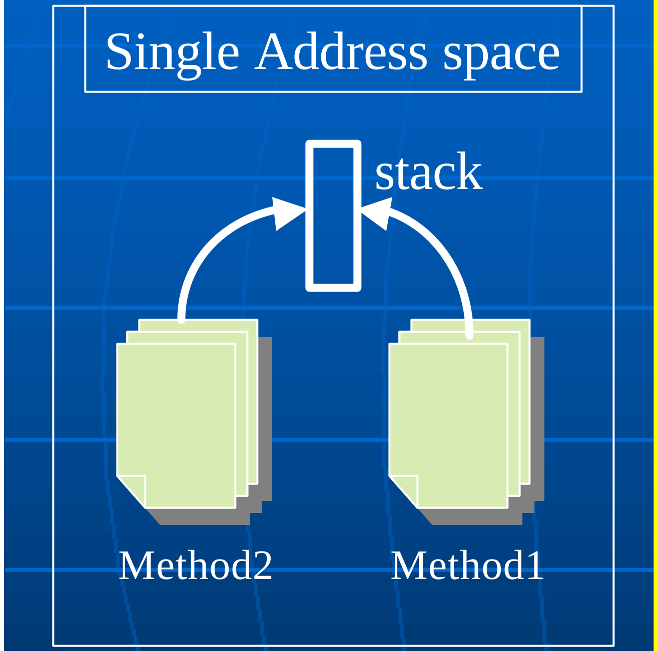


XML Defined Resources
connected by XML defined messages
Implementation of resource and
connection may or may not be XML

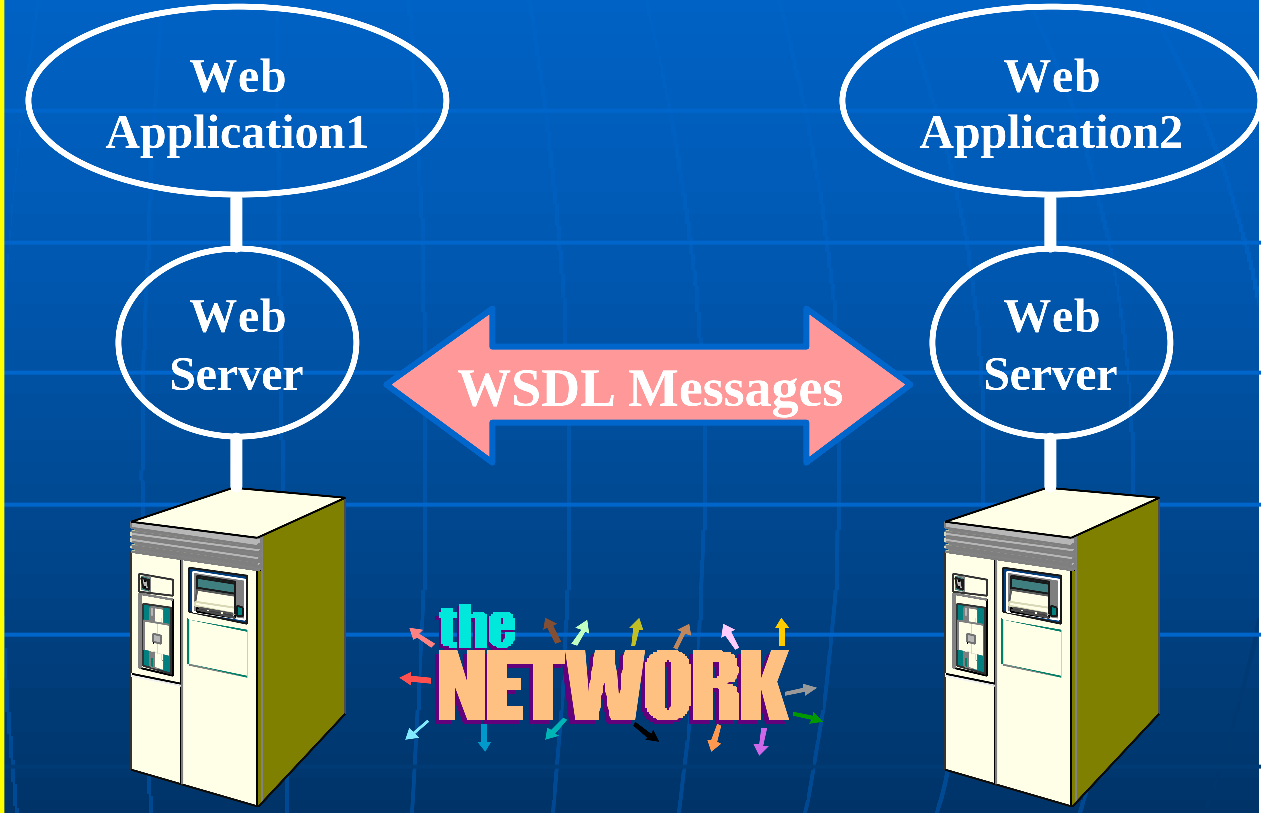
Some Research Issues for P2P Grid

- What happens to programming languages when data structures are defined in XML
- How do we manage a sea of virtual XML?
  - Register, find and link objects
  - This is “distributed operating system of the world” ?
- What happens to databases when everything is an Object defined in XML and transformed by Java?
- How and when do we compile virtual XML
  - Convert slow XML message to super fast method call on stack
- How do we implement services such as Security and collaboration over a range of grain sizes
- Supporting all “grain sizes” we get some sort of dynamic fractal world which looks like XML objects exchanging XML messages at all scales
  - Not well supported by centralized services (P2P problem)
- Semantic Grid: as metadata increases, objects link together forming digital brilliance – a phase transition in information space

Compiling for WSDL

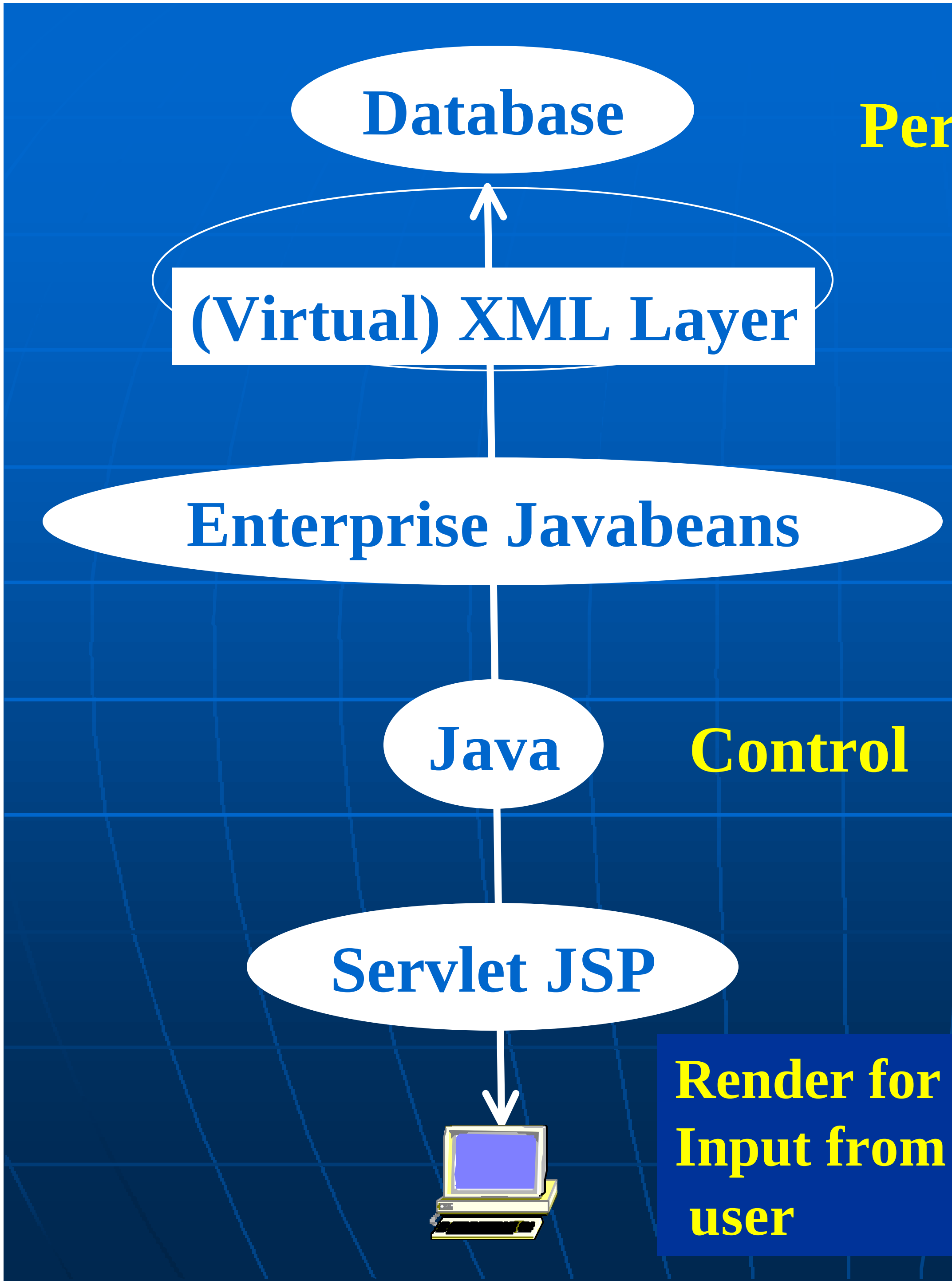


Shared Memory



Distributed System

.opennet Architecture



Database

Persistent Managed Store

(Virtual) XML Layer

Object layer

Enterprise Javabeans

Virtual Machine

Java

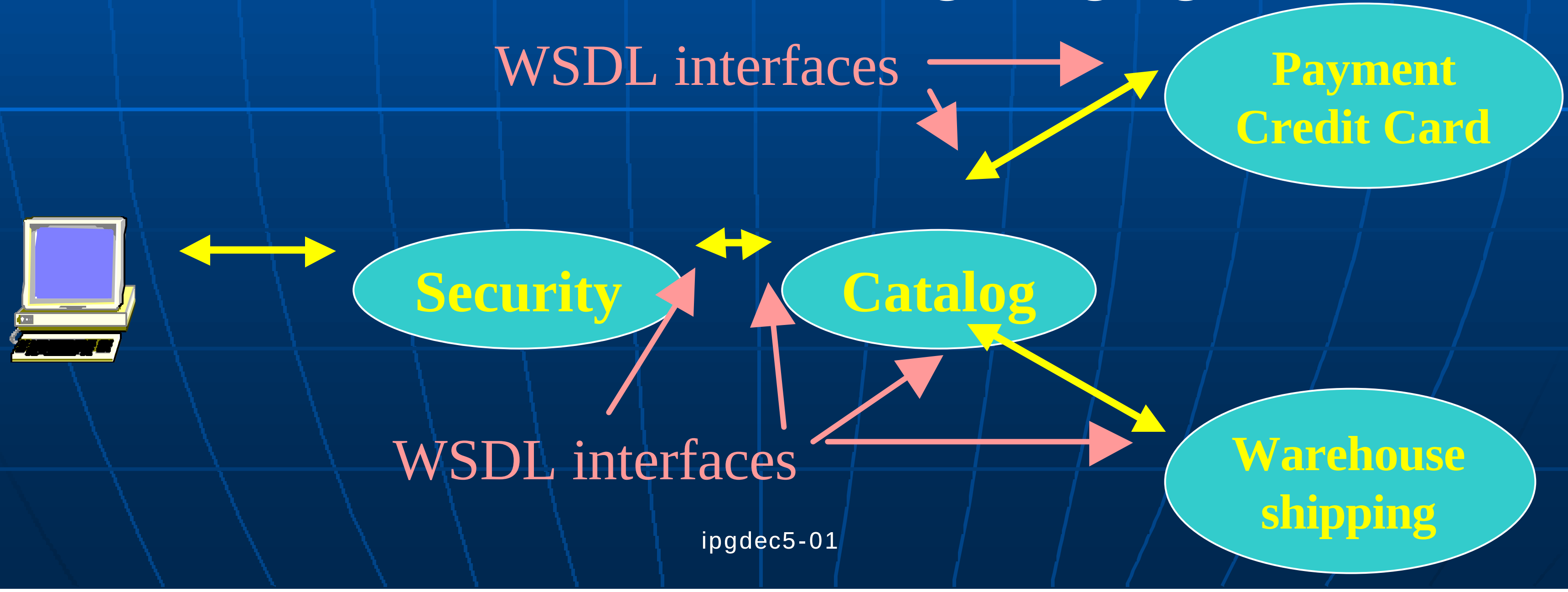
Control

Servlet JSP

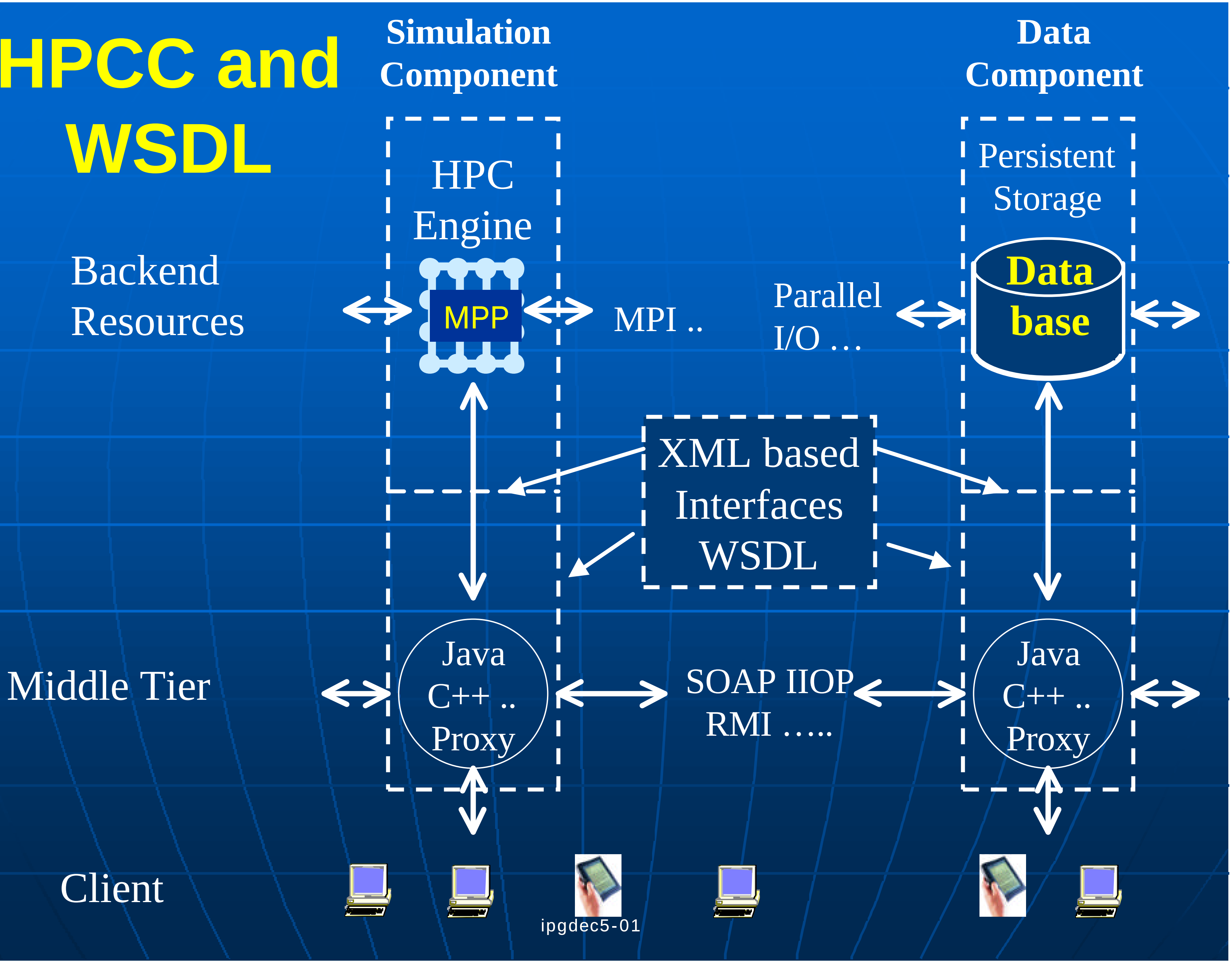
Render for
Input from
user

Role of Web Services

- Define interfaces of web applications so that computer-computer interactions are enabled
  - Defines virtual XML for all system and application services
- **WSDL** is XML versions of Class and Method definitions
- **SOAP** is XML version of message
- **UDDI** or **WSIL** catalogs WSDL based services enabling precise linkage of them
- **WSFL** and **WSCL** are candidate linkage languages



HPCC and WSDL



Simulation Component

Data Component

Backend Resources

HPC Engine

MPP

MPI ..

Parallel I/O …

Persistent Storage

Data base

XML based Interfaces WSDL

Middle Tier

Java C++ .. Proxy

SOAP IIOP RMI …..

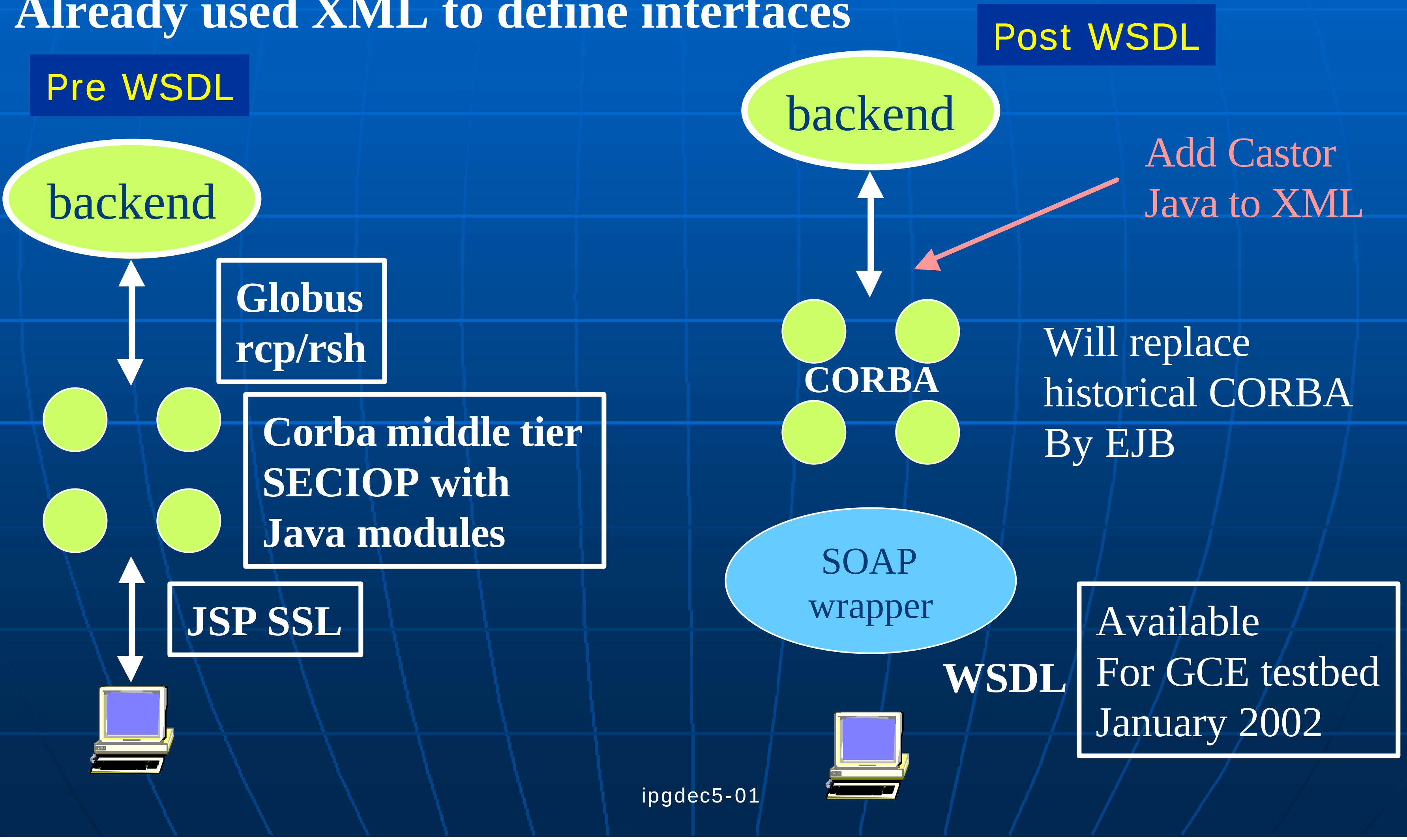
Java C++ .. Proxy

Client

ipgdec5-01

Converting a Portal to WSDL

- Gateway (http://www.gatewayportal.org) is a relatively mature portal supporting Job submission, management and some visualization for codes like ANSYS – developed for DoD HPC centers
- Already used XML to define interfaces



Pre WSDL

backend

Globus rcp/rsh

Corba middle tier SECIOP with Java modules

JSP SSL

Post WSDL

backend

Add Castor Java to XML

CORBA

Will replace historical CORBA By EJB

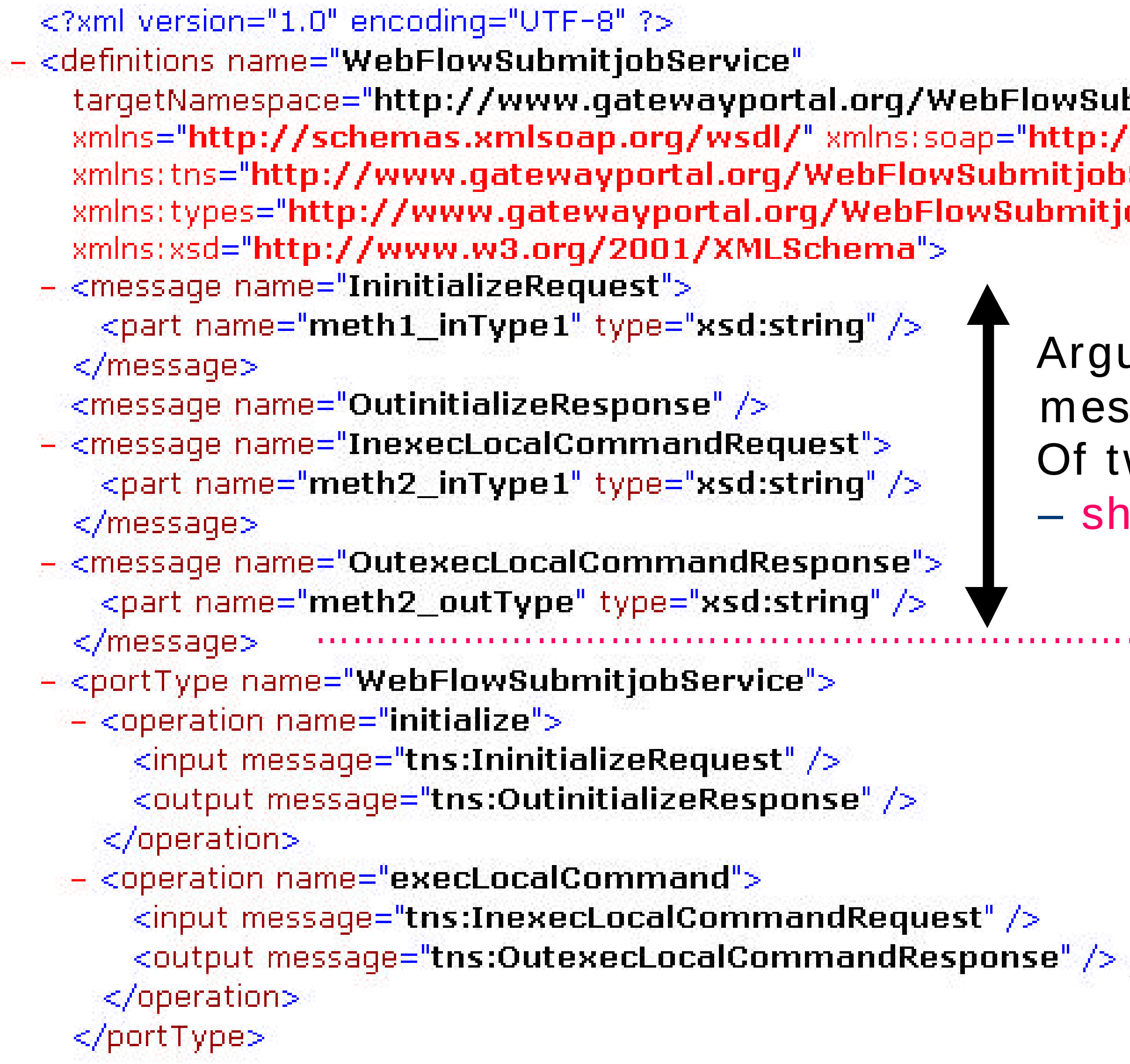
SOAP wrapper

WSDL

Available For GCE testbed January 2002

WSDL Job Submittal service I

- More details at http://www.gatewayportal.org/ and http://aspen.ucs.indiana.edu/ptliu/gatewaywsdl

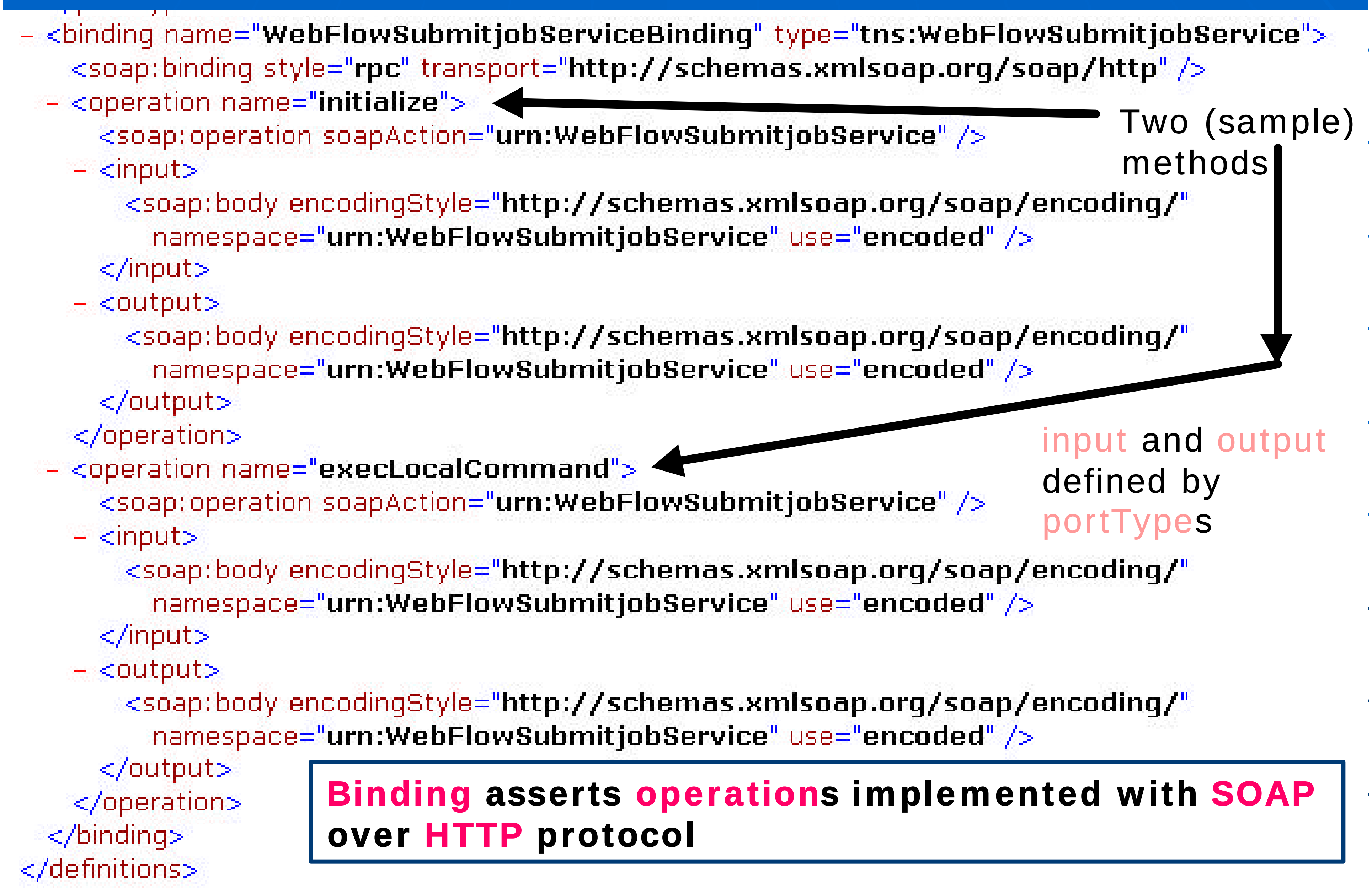


```
  <?xml version="1.0" encoding="UTF-8" ?>
- <definitions name="WebFlowSubmitjobService"
    targetNamespace="http://www.gatewayportal.org/WebFlowSubmitjobService-interface"
    xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tns="http://www.gatewayportal.org/WebFlowSubmitjobService-interface"
    xmlns:types="http://www.gatewayportal.org/WebFlowSubmitjobService-interface/types/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  - <message name="IninitializeRequest">
      <part name="meth1_inType1" type="xsd:string" />
    </message>
    <message name="OutinitializeResponse" />
  - <message name="InexecLocalCommandRequest">
      <part name="meth2_inType1" type="xsd:string" />
    </message>
  - <message name="OutexecLocalCommandResponse">
      <part name="meth2_outType" type="xsd:string" />
    </message>
  - <portType name="WebFlowSubmitjobService">
    - <operation name="initialize">
        <input message="tns:IninitializeRequest" />
        <output message="tns:OutinitializeResponse" />
      </operation>
    - <operation name="execLocalCommand">
        <input message="tns:InexecLocalCommandRequest" />
        <output message="tns:OutexecLocalCommandResponse" />
      </operation>
    </portType>
```

Arguments and return (as messages)
Of two RPC methods in Gateway
– should standardize

(abstract) portType without binding to Transport or Address
operation " method
Define RPC like methods with in and out parameters

WSDL Job Submittal service II



```
- <binding name="WebFlowSubmitjobServiceBinding" type="tns:WebFlowSubmitjobService">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
  - <operation name="initialize">
      <soap:operation soapAction="urn:WebFlowSubmitjobService" />
    - <input>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="urn:WebFlowSubmitjobService" use="encoded" />
      </input>
    - <output>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="urn:WebFlowSubmitjobService" use="encoded" />
      </output>
    </operation>
  - <operation name="execLocalCommand">
      <soap:operation soapAction="urn:WebFlowSubmitjobService" />
    - <input>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="urn:WebFlowSubmitjobService" use="encoded" />
      </input>
    - <output>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="urn:WebFlowSubmitjobService" use="encoded" />
      </output>
    </operation>
  </binding>
</definitions>
```

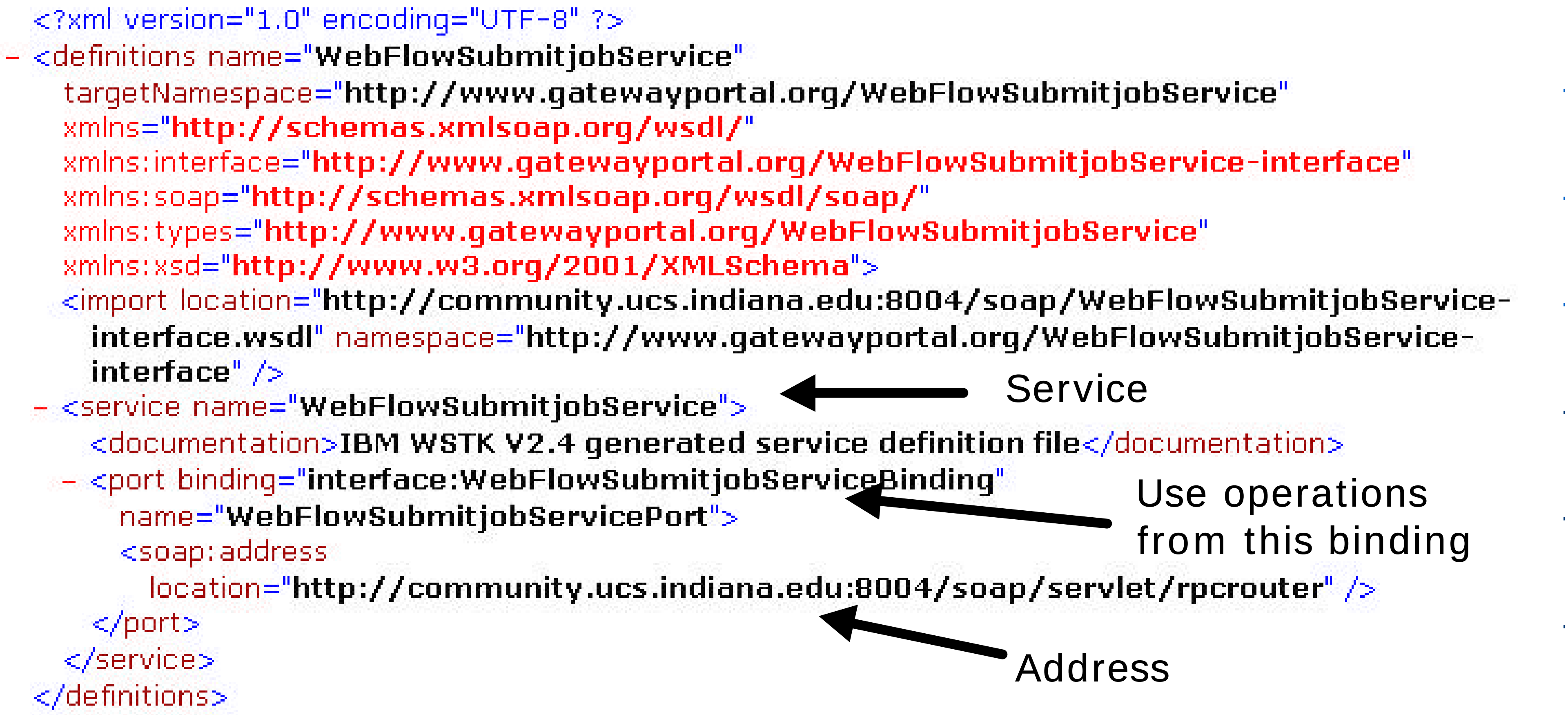
Two (sample) methods

input and output defined by portTypes

Binding asserts operations implemented with SOAP over HTTP protocol

WSDL Job Submittal service III

- Define WebFlowSubmitjobService with a single port implementing previous binding at a particular port
- Uses WSDL import syntax to reference previous specifications



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <definitions name="WebFlowSubmitjobService"
    targetNamespace="http://www.gatewayportal.org/WebFlowSubmitjobService"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:interface="http://www.gatewayportal.org/WebFlowSubmitjobService-interface"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:types="http://www.gatewayportal.org/WebFlowSubmitjobService"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <import location="http://community.ucs.indiana.edu:8004/soap/WebFlowSubmitjobService-
      interface.wsdl" namespace="http://www.gatewayportal.org/WebFlowSubmitjobService-
      interface" />
  - <service name="WebFlowSubmitjobService">
      <documentation>IBM WSTK V2.4 generated service definition file</documentation>
    - <port binding="interface:WebFlowSubmitjobServiceBinding"
        name="WebFlowSubmitjobServicePort">
        <soap:address
          location="http://community.ucs.indiana.edu:8004/soap/servlet/rpcrouter" />
      </port>
    </service>
  </definitions>
```

Service

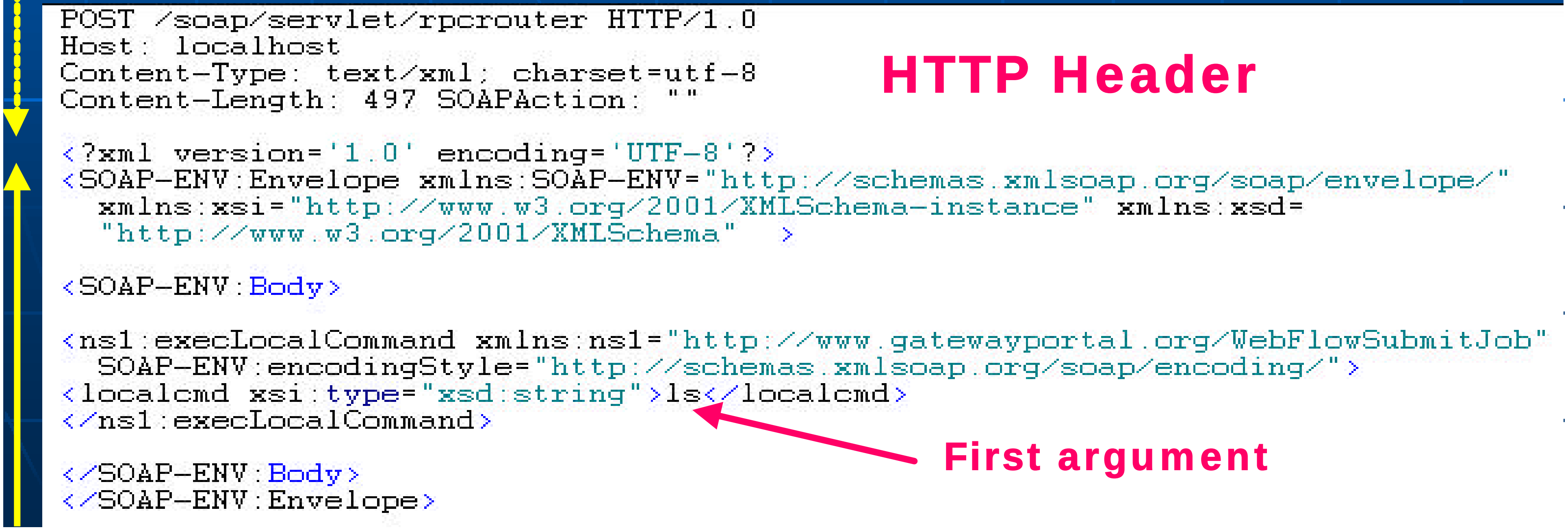
Use operations from this binding

Address

SOAP and Gateway Portal I

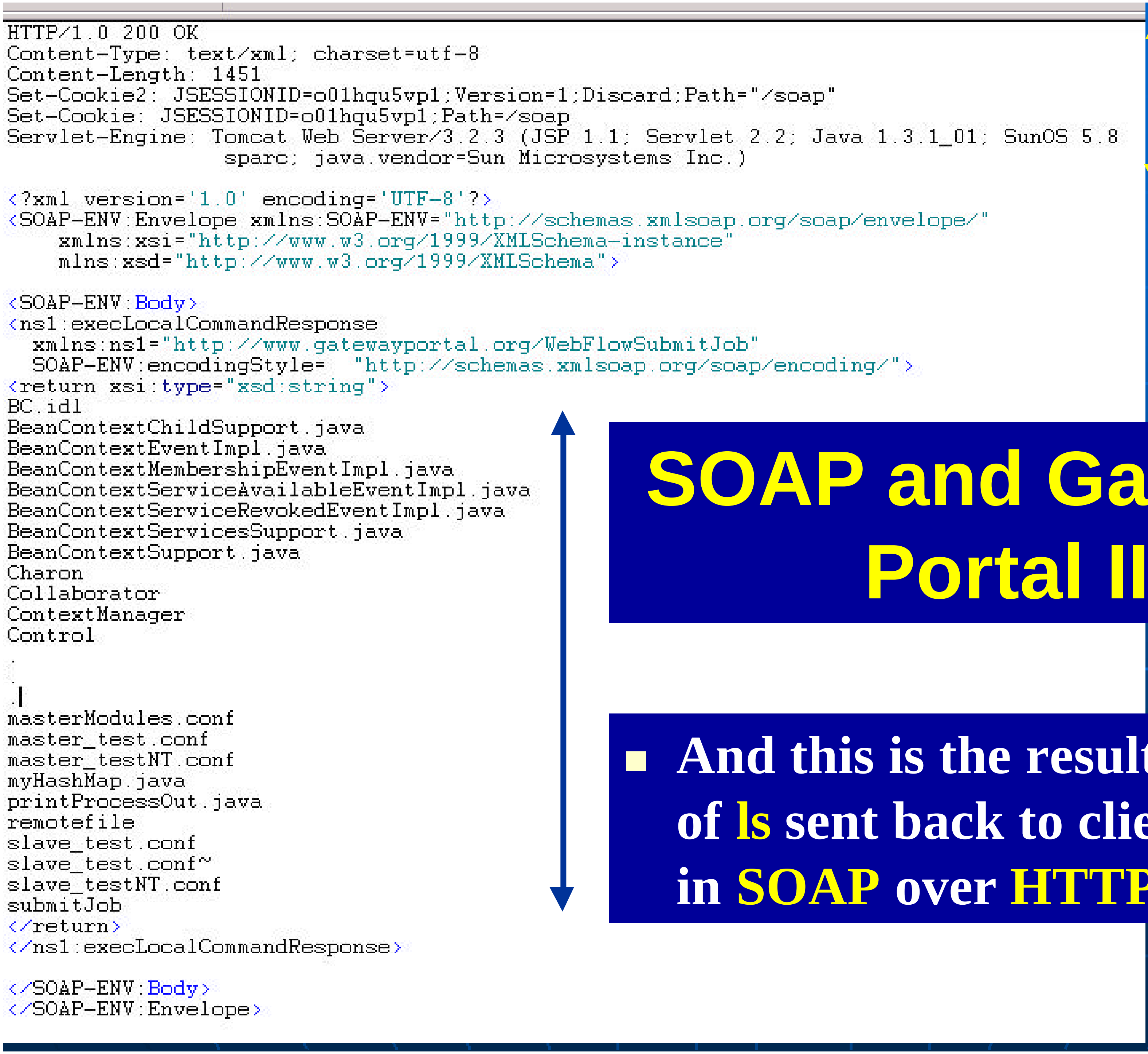
- Having specified service in WSDL, the run-time is implemented in SOAP
- Here is SOAP over HTTP message from client
- This is execLocalCommand to run one particular command (ls) on current WebFlow directory

SOAP
Envelope
With body



```
POST /soap/servlet/rpcrouter HTTP/1.0
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: 497 SOAPAction: ""

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd=
  "http://www.w3.org/2001/XMLSchema"  >

<SOAP-ENV:Body>

<ns1:execLocalCommand xmlns:ns1="http://www.gatewayportal.org/WebFlowSubmitJob"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<localcmd xsi:type="xsd:string">ls</localcmd>
</ns1:execLocalCommand>

</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP and Gateway Portal II



```
HTTP/1.0 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 1451
Set-Cookie2: JSESSIONID=o01hqu5vp1;Version=1;Discard;Path="/soap"
Set-Cookie: JSESSIONID=o01hqu5vp1;Path=/soap
Servlet-Engine: Tomcat Web Server/3.2.3 (JSP 1.1; Servlet 2.2; Java 1.3.1_01; SunOS 5.8
                 sparc; java.vendor=Sun Microsystems Inc.)

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
    mlns:xsd="http://www.w3.org/1999/XMLSchema">

<SOAP-ENV:Body>
<ns1:execLocalCommandResponse
  xmlns:ns1="http://www.gatewayportal.org/WebFlowSubmitJob"
  SOAP-ENV:encodingStyle=  "http://schemas.xmlsoap.org/soap/encoding/">
<return xsi:type="xsd:string">
BC.idl
BeanContextChildSupport.java
BeanContextEventImpl.java
BeanContextMembershipEventImpl.java
BeanContextServiceAvailableEventImpl.java
BeanContextServiceRevokedEventImpl.java
BeanContextServicesSupport.java
BeanContextSupport.java
Charon
Collaborator
ContextManager
Control
.
.
.
masterModules.conf
master_test.conf
master_testNT.conf
myHashMap.java
printProcessOut.java
remotefile
slave_test.conf
slave_test.conf~
slave_testNT.conf
submitJob
</return>
</ns1:execLocalCommandResponse>

</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

HTTP Header

SOAP Envelope and body

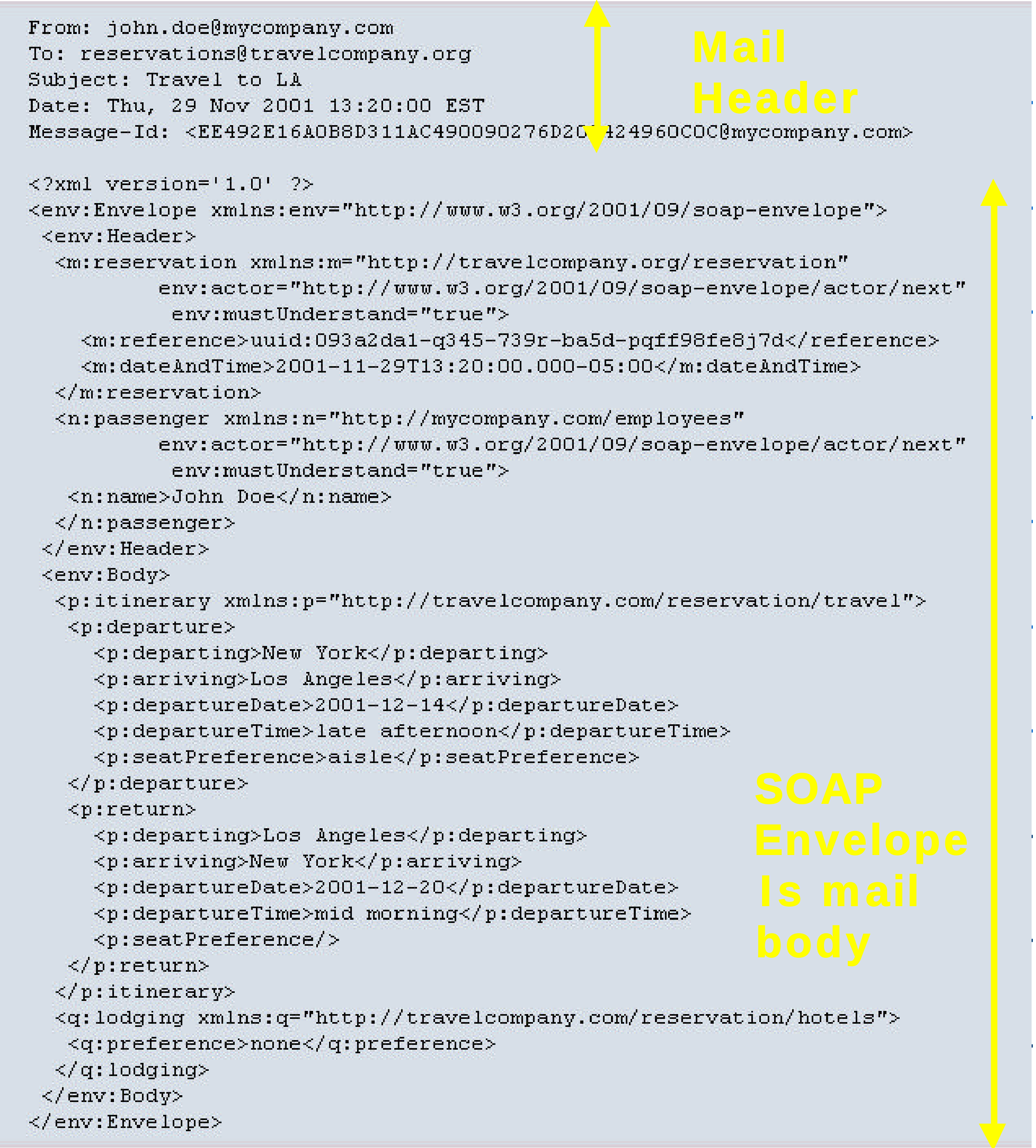
- And this is the result of ls sent back to client in SOAP over HTTP

Next Steps in WSDL Portals

- Agree on WSDL Interfaces for important job submittal and management functions
  - Are computers also defined in WSDL – believe so
- Set up UDDI servers to catalog amnd retrieve WSDL services
  - How is this consistent with current Grid Information Services?
- Set up interoperability test bed
- Build "HPCC compiled" web services
- Look at other computational science applications
  - Databases
  - NASA/EU SLE (Space Link extension) standard for ground stations for sensors

SOAP Binding to SMTP

- You can use this to queue up your job requests by email on your airtrip and send when you land
- Value of separation of function and protocol



```
From: john.doe@mycompany.com
To: reservations@travelcompany.org
Subject: Travel to LA
Date: Thu, 29 Nov 2001 13:20:00 EST
Message-Id: <EE492E16A0B8D311AC490090276D208424960C0C@mycompany.com>

<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2001/09/soap-envelope">
 <env:Header>
  <m:reservation xmlns:m="http://travelcompany.org/reservation"
          env:actor="http://www.w3.org/2001/09/soap-envelope/actor/next"
           env:mustUnderstand="true">
    <m:reference>uuid:093a2da1-q345-739r-ba5d-pqff98fe8j7d</reference>
    <m:dateAndTime>2001-11-29T13:20:00.000-05:00</m:dateAndTime>
  </m:reservation>
  <n:passenger xmlns:n="http://mycompany.com/employees"
          env:actor="http://www.w3.org/2001/09/soap-envelope/actor/next"
           env:mustUnderstand="true">
   <n:name>John Doe</n:name>
  </n:passenger>
 </env:Header>
 <env:Body>
  <p:itinerary xmlns:p="http://travelcompany.com/reservation/travel">
   <p:departure>
     <p:departing>New York</p:departing>
     <p:arriving>Los Angeles</p:arriving>
     <p:departureDate>2001-12-14</p:departureDate>
     <p:departureTime>late afternoon</p:departureTime>
     <p:seatPreference>aisle</p:seatPreference>
   </p:departure>
   <p:return>
     <p:departing>Los Angeles</p:departing>
     <p:arriving>New York</p:arriving>
     <p:departureDate>2001-12-20</p:departureDate>
     <p:departureTime>mid morning</p:departureTime>
     <p:seatPreference/>
   </p:return>
  </p:itinerary>
  <q:lodging xmlns:q="http://travelcompany.com/reservation/hotels">
   <q:preference>none</q:preference>
  </q:lodging>
 </env:Body>
</env:Envelope>
```

Mail Header

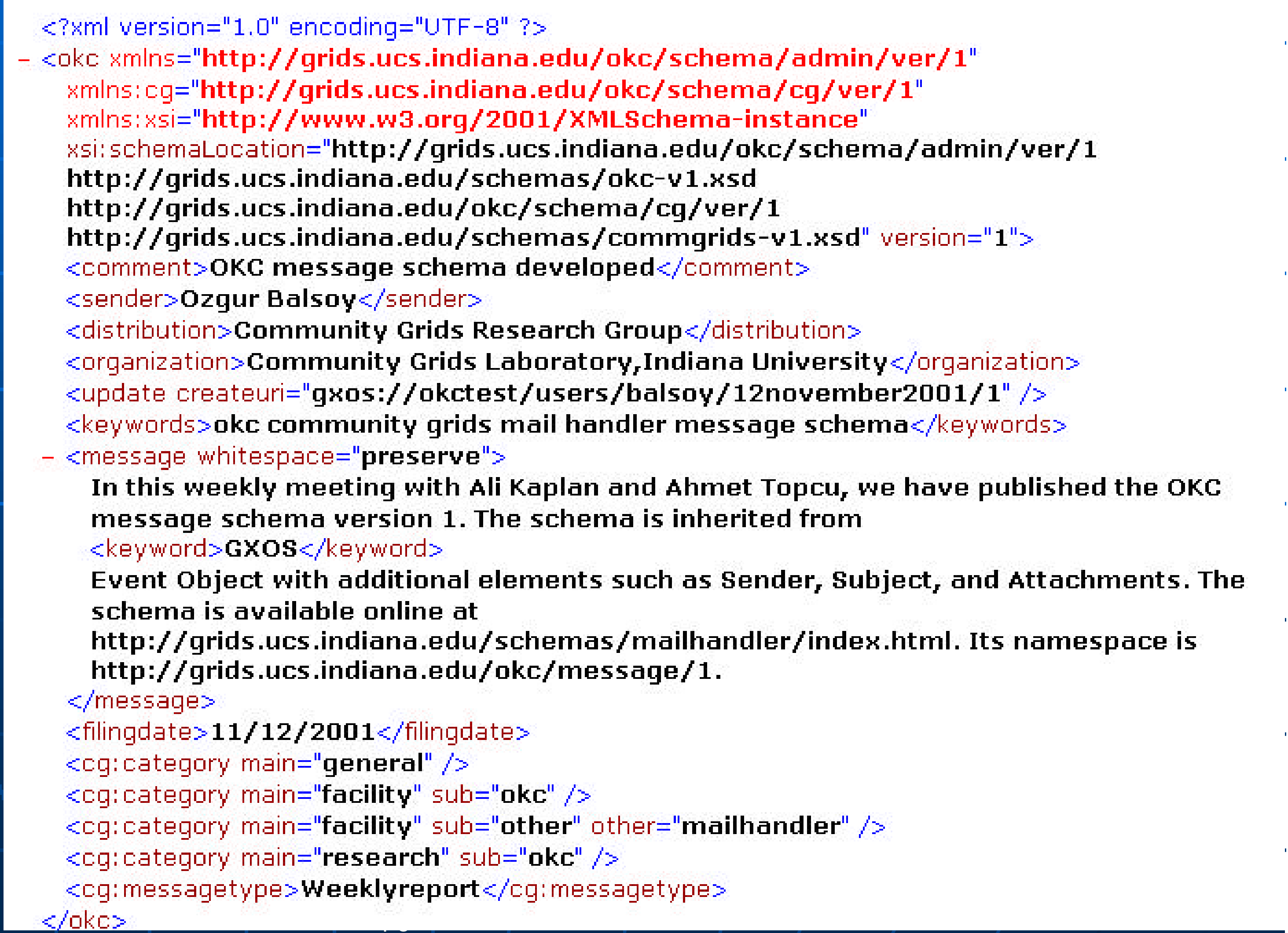
SOAP Envelope Is mail body

Threaded Discussion/Reporting as a Web Service

- Support email or form based reporting/discussion

Design an Application Specific Schema
Can of course process email as Web service

Testing for Student reports And Web site updates with report Web Service built around “publish/ subscribe” Web Service (later)



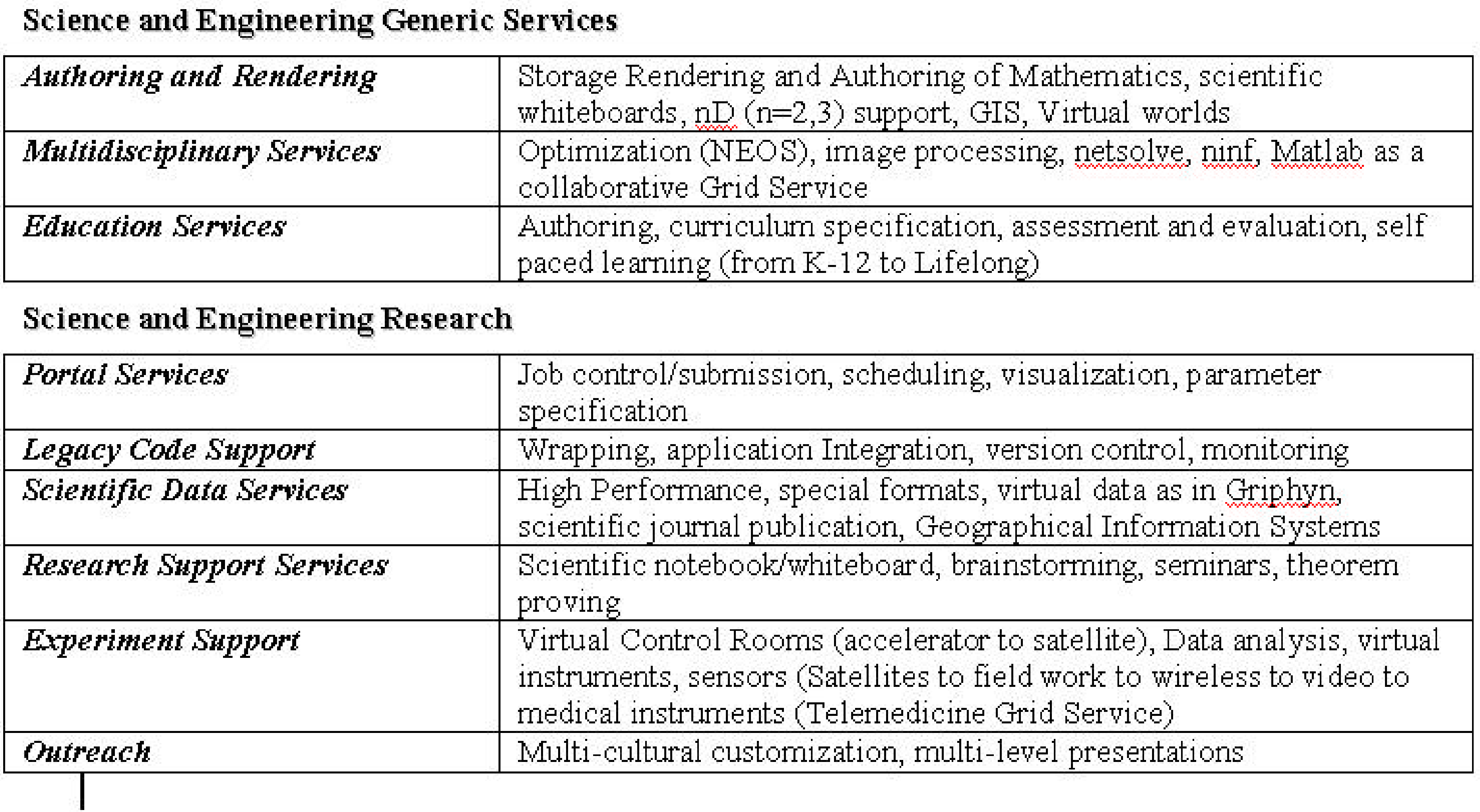
```
  <?xml version="1.0" encoding="UTF-8" ?>
- <okc xmlns="http://grids.ucs.indiana.edu/okc/schema/admin/ver/1"
    xmlns:cg="http://grids.ucs.indiana.edu/okc/schema/cg/ver/1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://grids.ucs.indiana.edu/okc/schema/admin/ver/1
    http://grids.ucs.indiana.edu/schemas/okc-v1.xsd
    http://grids.ucs.indiana.edu/okc/schema/cg/ver/1
    http://grids.ucs.indiana.edu/schemas/commgrids-v1.xsd" version="1">
    <comment>OKC message schema developed</comment>
    <sender>Ozgur Balsoy</sender>
    <distribution>Community Grids Research Group</distribution>
    <organization>Community Grids Laboratory,Indiana University</organization>
    <update createuri="gxos://okctest/users/balsoy/12november2001/1" />
    <keywords>okc community grids mail handler message schema</keywords>
  - <message whitespace="preserve">
      In this weekly meeting with Ali Kaplan and Ahmet Topcu, we have published the OKC
      message schema version 1. The schema is inherited from
      <keyword>GXOS</keyword>
      Event Object with additional elements such as Sender, Subject, and Attachments. The
      schema is available online at
      http://grids.ucs.indiana.edu/schemas/mailhandler/index.html. Its namespace is
      http://grids.ucs.indiana.edu/okc/message/1.
    </message>
    <filingdate>11/12/2001</filingdate>
    <cg:category main="general" />
    <cg:category main="facility" sub="okc" />
    <cg:category main="facility" sub="other" other="mailhandler" />
    <cg:category main="research" sub="okc" />
    <cg:messagetype>Weeklyreport</cg:messagetype>
  </okc>
```

Science as a Web Service

- Build a network of linked web-based applications to support science
  - Simulation, visualization, data-input, data analysis, publication are web services made up themselves of smaller web services (like ls in Gateway!)
- Enable "plug and play" of modules so supporting "Science for the Americas"
  - Modules can vary from high end research to K-12 instruction
  - Enable a distributed less monolithic approach to research
  - People in network as research colleagues or mentors
- Requires collaborative web services

Some Science Web Services

- These build on general (community) web services



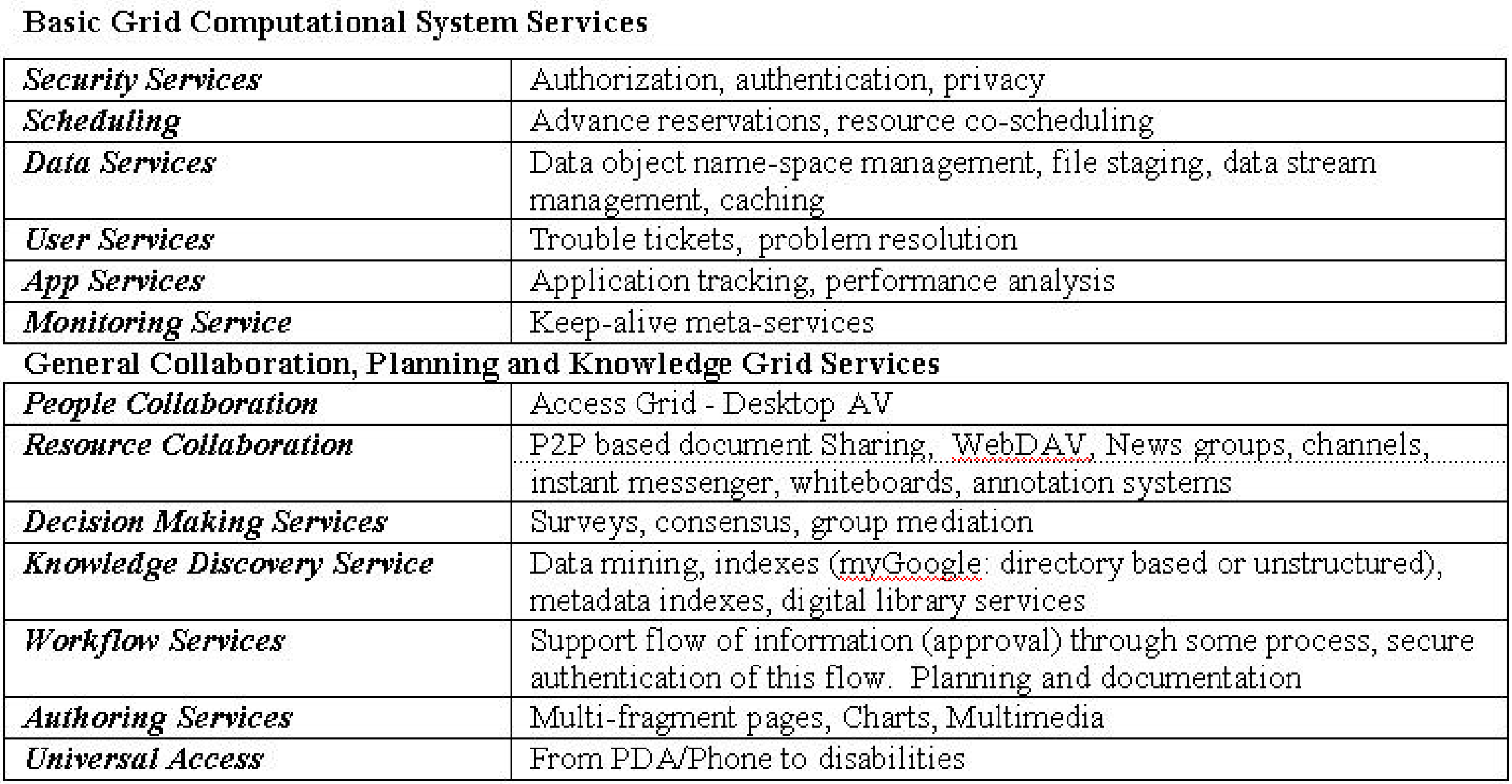
Science and Engineering Generic Services

Authoring and Rendering	Storage Rendering and Authoring of Mathematics, scientific whiteboards, nD (n=2,3) support, GIS, Virtual worlds
Multidisciplinary Services	Optimization (NEOS), image processing, netsolve, ninf, Matlab as a collaborative Grid Service
Education Services	Authoring, curriculum specification, assessment and evaluation, self paced learning (from K-12 to Lifelong)

Science and Engineering Research

Portal Services	Job control/submission, scheduling, visualization, parameter specification
Legacy Code Support	Wrapping, application Integration, version control, monitoring
Scientific Data Services	High Performance, special formats, virtual data as in Griphyn, scientific journal publication, Geographical Information Systems
Research Support Services	Scientific notebook/whiteboard, brainstorming, seminars, theorem proving
Experiment Support	Virtual Control Rooms (accelerator to satellite), Data analysis, virtual instruments, sensors (Satellites to field work to wireless to video to medical instruments (Telemedicine Grid Service)
Outreach	Multi-cultural customization, multi-level presentations

Some General Grid Web Services



Basic Grid Computational System Services

Security Services	Authorization, authentication, privacy
Scheduling	Advance reservations, resource co-scheduling
Data Services	Data object name-space management, file staging, data stream management, caching
User Services	Trouble tickets, problem resolution
App Services	Application tracking, performance analysis
Monitoring Service	Keep-alive meta-services

General Collaboration, Planning and Knowledge Grid Services

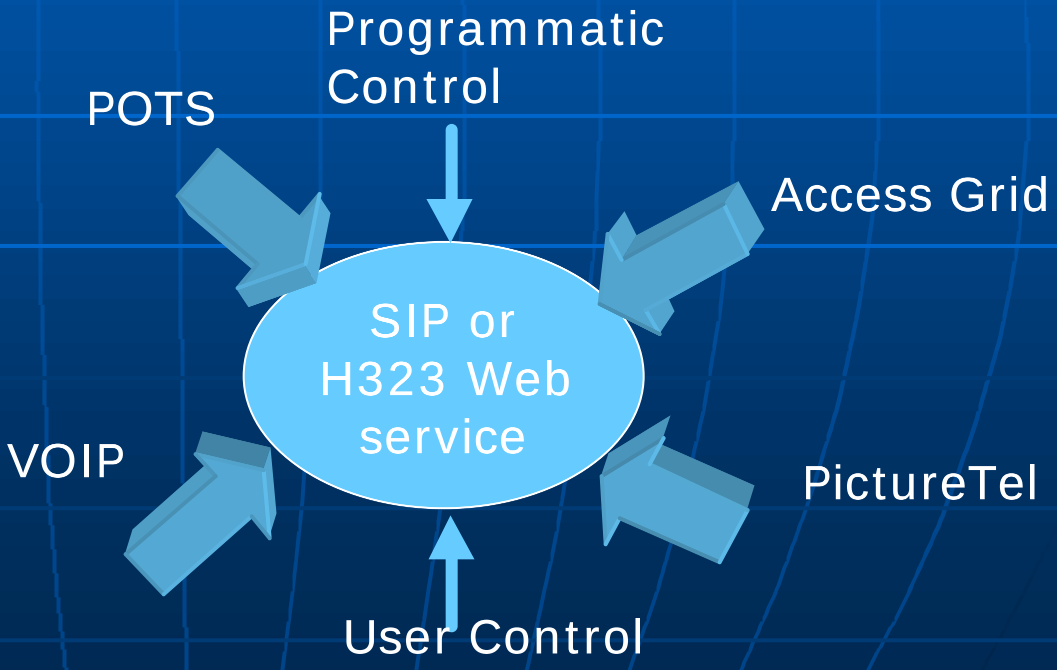
People Collaboration	Access Grid - Desktop AV
Resource Collaboration	P2P based document Sharing, WebDAV, News groups, channels, instant messenger, whiteboards, annotation systems
Decision Making Services	Surveys, consensus, group mediation
Knowledge Discovery Service	Data mining, indexes (myGoogle: directory based or unstructured), metadata indexes, digital library services
Workflow Services	Support flow of information (approval) through some process, secure authentication of this flow. Planning and documentation
Authoring Services	Multi-fragment pages, Charts, Multimedia
Universal Access	From PDA/Phone to disabilities

Education as a Web Service

- Can link to Science as a Web Service and substitute educational modules
- "Learning Object" XML standards already exist from IMS/ADL http://www.adlnet.org – need to update architecture
- Web Services for virtual university include:
- Registration
- Performance (grading)
- Authoring of Curriculum
- Online laboratories for real and virtual instruments
- Homework submission
- Quizzes of various types (multiple choice, random parameters)
- Assessment data access and analysis
- Synchronous Delivery of Curricula
- Scheduling of courses and mentoring sessions
- Asynchronous access, data-mining and knowledge discovery

Audio Video Conferencing as a Web Service

- **This could be similar to vrvs.org with different ports corresponding to different protocols**
- **Use "universal messaging subsystem" to transmit A/V streams between sources and sinks**



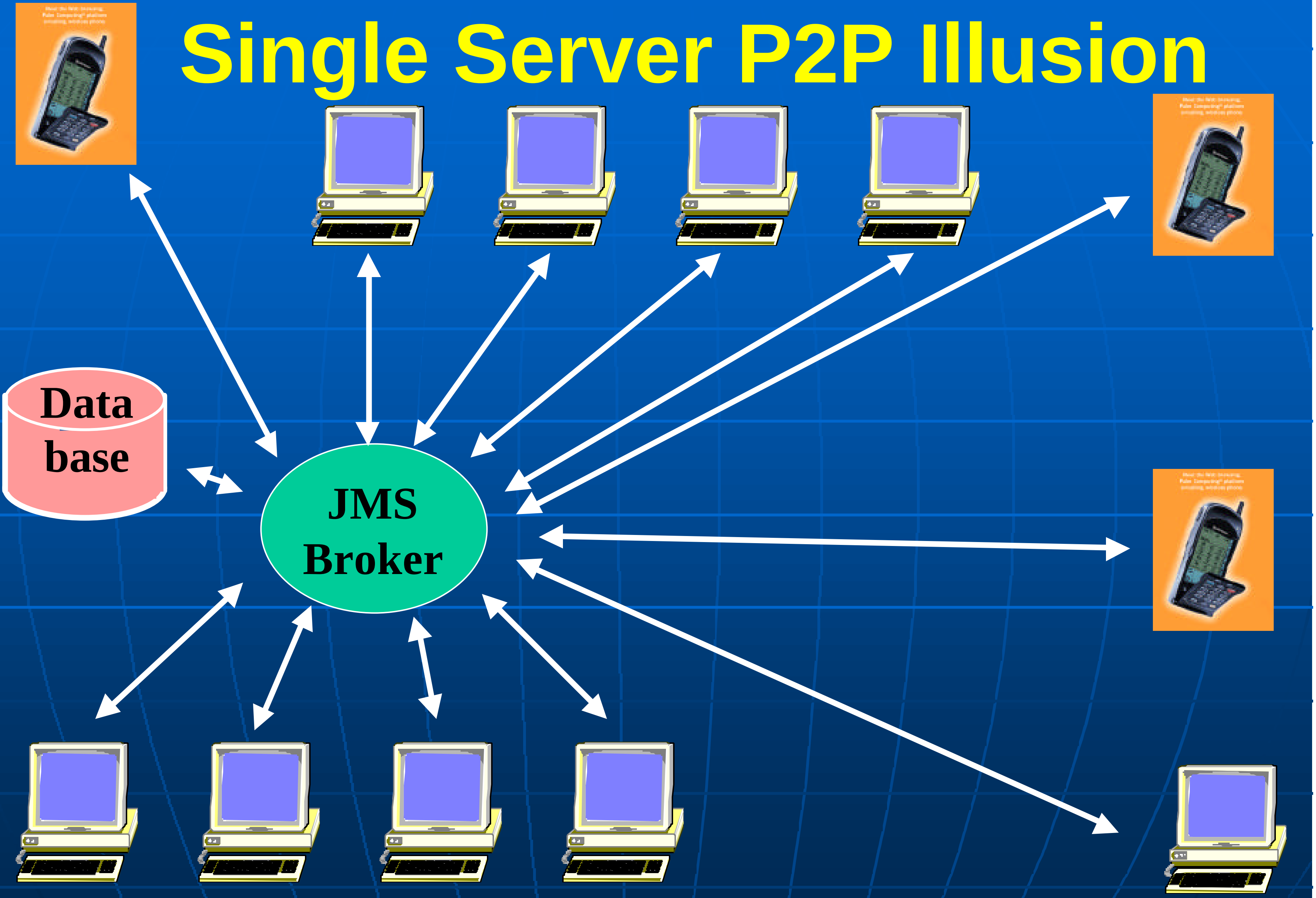
Semantic Grid & Digital Brilliance

- Peer to Peer networks teach us that we can build "small worlds" where distance between nodes is logarithmic in number of nodes
- Consider a Grid of WSDL services linked (through UDDI) together
  - This is spirit of Semantic web – metadata enables meaningful linkage
- We do not need to link everybody but only to establish "small world" routes
- Physics analogies suggest that phase transitions will occur when "enough" nodes are linked – one will get nodes to align in the direction of new knowledge
- This suggests ways of quantifying value of metadata induced linkages and areas where one "should" add more WSDL specifications

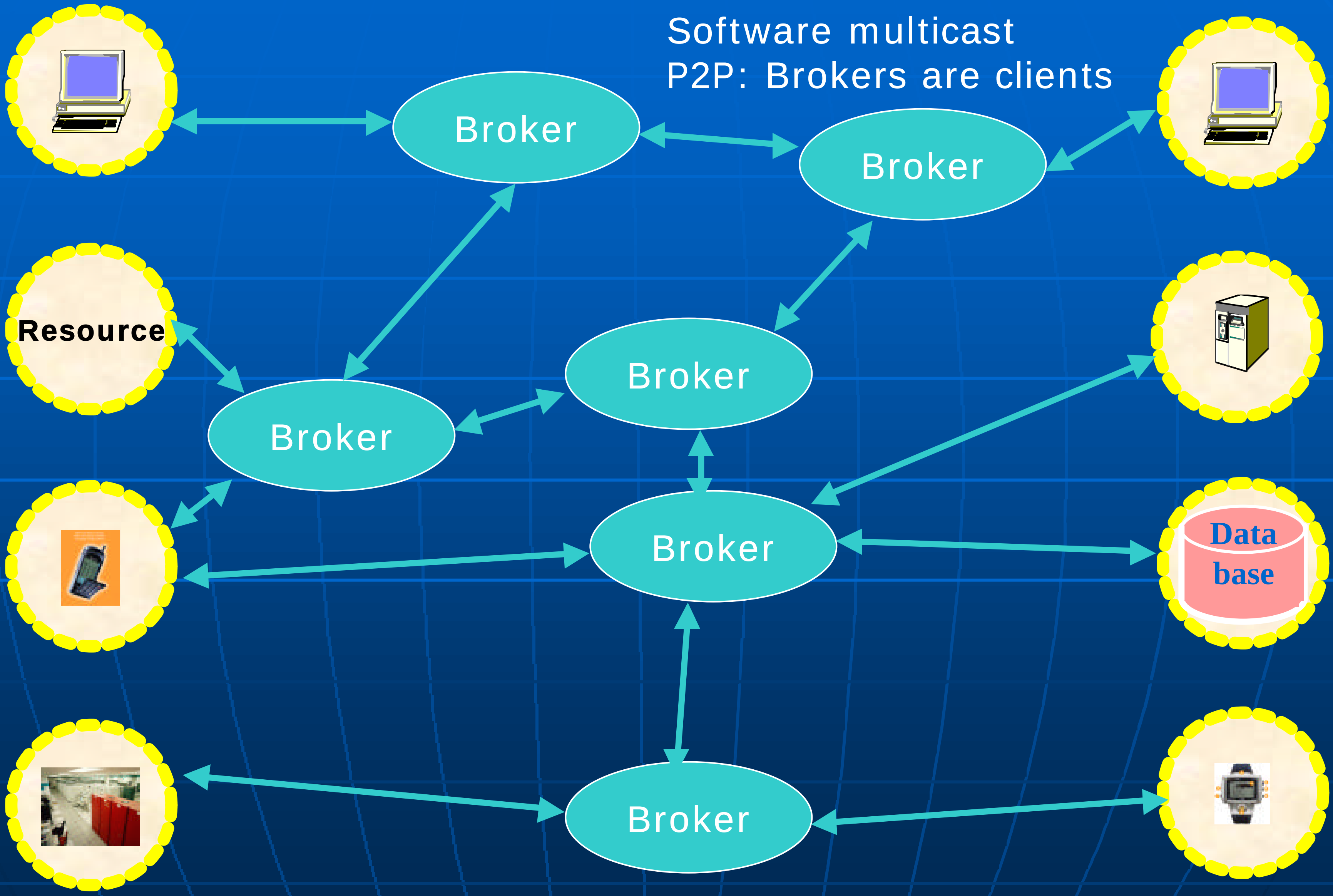
Publish/Subscribe as a Web Service

- We can implement messaging subsystem (between WSDL resources) with either direct messages or by a queued system where you publish messages to queues and subscribe as receiver to particular queues
  - Natural asynchronous collaboration model which is in fact fast enough for synchronous collaboration
- There are many different publish/subscribe models
  - JMS is a cluster of central servers
  - JXTA is a very dynamic Peer to Peer model where pipes are queues and topics (metadata) are service advertisements
- Implement JMS API with JXTA protocol – different WSDL bindings here have different fault tolerance/reliability semantics
  - Could use JMS as long distance "carrier" between JXTA peers
  - JXTA provides higher performance than JMS for nearby recipients
- Pallickara built an intermediate dynamic GXOS message broker subsystem

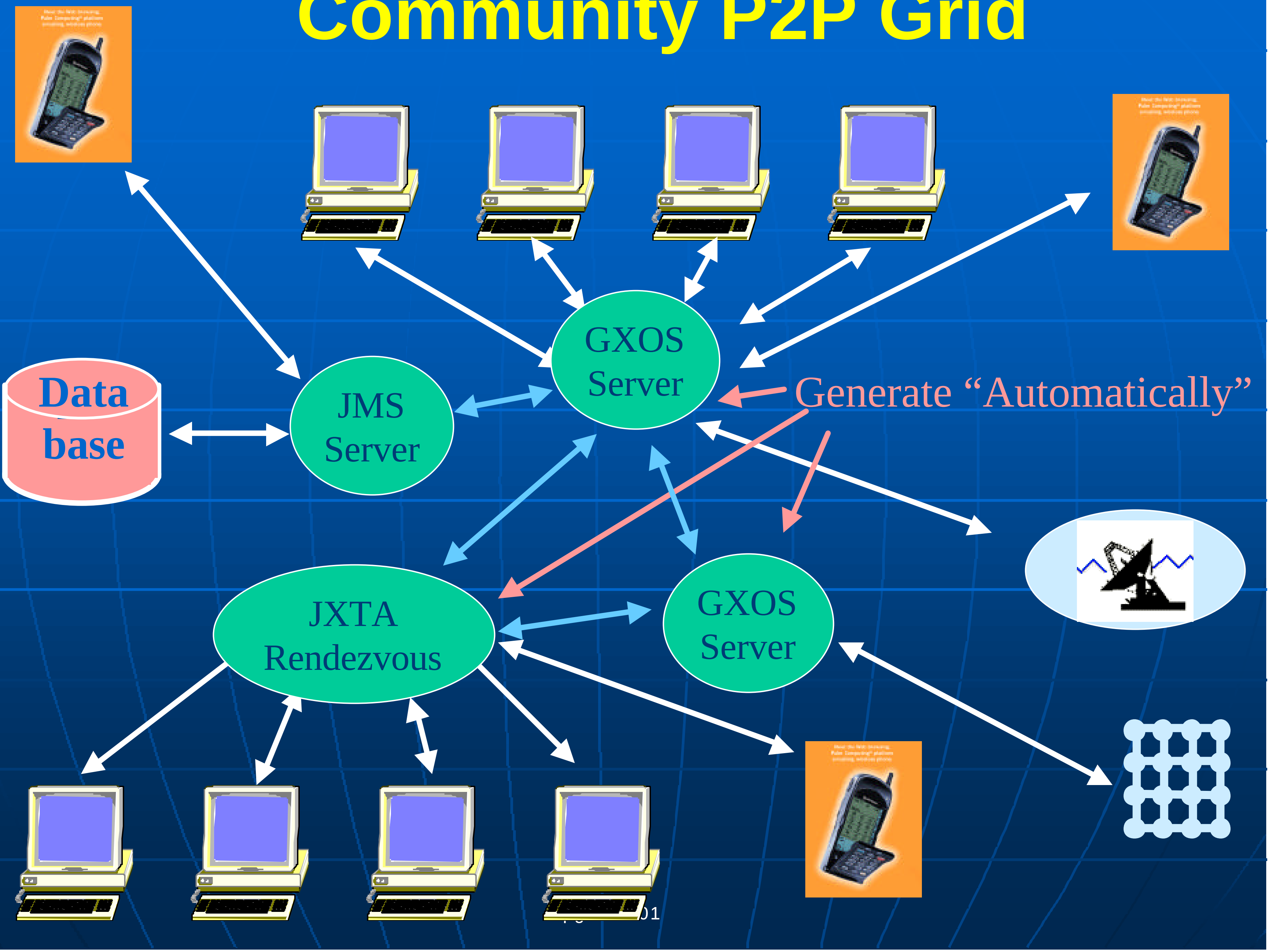
Single Server P2P Illusion



Broker Network


Software multicast
P2P: Brokers are clients

Community P2P Grid



Collaborative Web Resources

- Collaboration is “just” sharing objects
- What about Collaborative Web Services ?
  - You can in some cases do this automatically just by multicasting messages from service to clients
  - This is achieved by service publishing messages and clients subscribing
- Many applications do not expose all state changes
  - E.g. when I edit PowerPoint slide, PowerPoint does not tell the world by sending an (XML) message
- Solved by shared event collaboration model and requires one to view user interface as a “port” in WSDL sense and treat “event handlers” (mouseover, click etc.) as messages in WSDL
- Groove Networks does use XML front end to COM interfaces
  - More elegantly can use W3C DOM for (the few) documents (SVG is one) and “universal event handlers”
- Interesting research area